Introduction to Embedded Systems

EECS 124, UC Berkeley, Spring 2008
Lecture 23: Localization and Mapping

Gabe Hoffmann
Ph.D. Candidate, Aero/Astro Engineering
Stanford University

Overview

- Statistical Models
- Localization
- Occupancy Grid Mapping
- Simultaneous Localization and Mapping (SLAM)

This lecture draws material from
S. Thrun, W. Burgard, & D. Fox, *Probabilistic Robotics*, 2005

Statistics Background

- Conditional Probability

$$p(x,z) = p(x|z)p(z) = p(z|x)p(x)$$

- Bayes' Rule

$$p(x|z) = \frac{p(z|x)p(x)}{p(z)}$$

- Independence of two variables

$$p(x,y) = p(x)p(y)$$

$$p(x|y) = \frac{p(x,y)}{p(y)} = \frac{p(x)p(y)}{p(y)} = p(x)$$

Conditional Independence

Statistics Background (cont.)

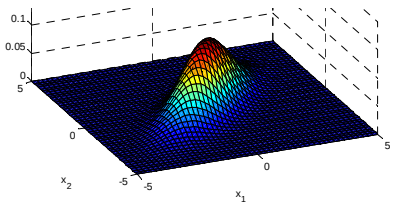
- Theorem of Total Probability

$$p(x) = \sum_y p(x,y) = \sum_y p(x|y)p(y)$$

$$p(x) = \int_y p(x,y)dy = \int_y p(x|y)p(y)dy$$

- Gaussian Probability Distributions

$$p(x) = \frac{1}{(2\pi)^{N/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-}1(x-\mu)\right)$$

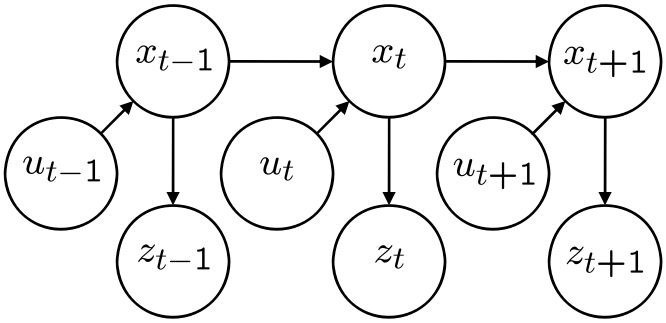


$$x \sim \mathcal{N}(\mu, \Sigma)$$

$$Ax \sim \mathcal{N}(A\mu, A\Sigma A^T)$$

Markov Assumption

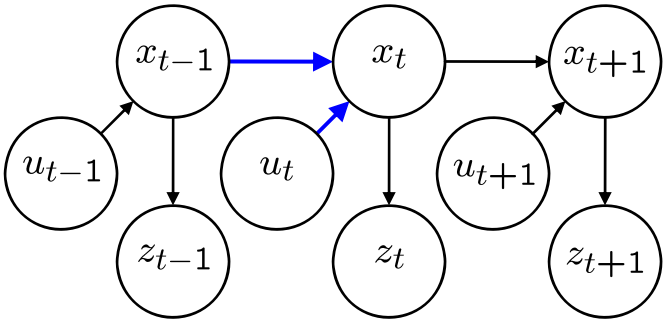
- Future and past independent given present



$$p(x_t|u_{1:t}, x_{0:t-1}) = p(x_t|u_t, x_{t-1})$$
$$p(z_t|u_{1:t}, x_{0:t}, z_{1:t-1}) = p(z_t|x_t)$$

Motion Model

- Stochastic model of future state: $p(x_t|u_t, x_{t-1})$

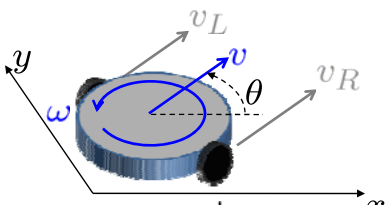


Two Wheeled Robots

- Velocity input model

Command Noise

$$\begin{aligned} v_L &= v_{L,c} + \xi_L \\ v_R &= v_{R,c} + \xi_R \end{aligned}$$



- Odometry "input" model

Measurement Noise

$$\begin{aligned} v &= d_m/\Delta t + \xi_v \\ \omega &= \theta_m/\Delta t + \xi_\omega \end{aligned}$$

$$\begin{aligned} v &= \frac{v_R + v_L}{2} \\ \omega &= \frac{v_R - v_L}{w} \end{aligned}$$

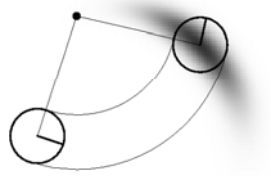
Two Wheeled Robots (cont.)

- Next state

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t + \frac{v_t}{\omega_t}(\sin(\theta_t + \omega_t \Delta t) - \sin\theta_t) \\ y_t + \frac{v_t}{\omega_t}(-\cos(\theta_t + \omega_t \Delta t) + \cos\theta_t) \\ \theta_t + \omega_t \Delta t \end{bmatrix}$$

- Noise model

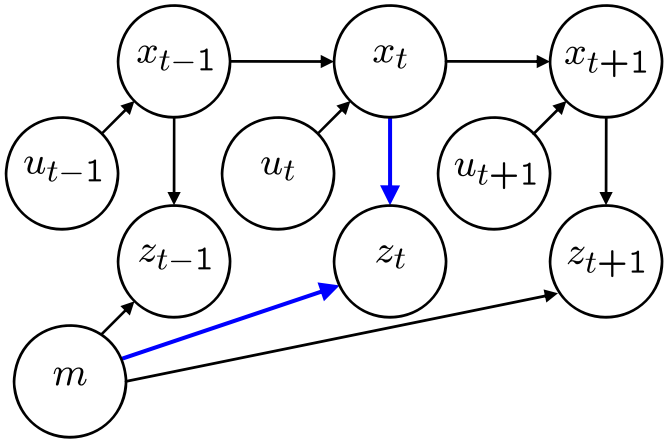
$$\begin{bmatrix} \xi_v \\ \xi_\omega \end{bmatrix} \sim \mathcal{N}(\vec{0}, M_t)$$



$$M_t = \begin{bmatrix} (\alpha_1|v_t| + \alpha_2|\omega_t|)^2 & 0 \\ 0 & (\alpha_3|v_t| + \alpha_4|\omega_t|)^2 \end{bmatrix}$$

(Forward) Measurement Model

- Stochastic model of measurements: $p(z_t|x_t, m)$

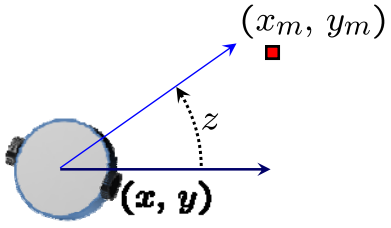


Sensor Examples

- Bump
- GPS
- Camera
- Directional Antenna
- Ultrasonic Ranger
- LIDAR

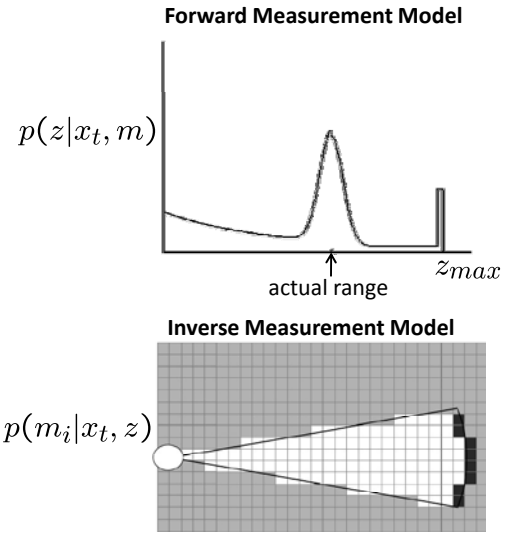
Bearing Sensor Model

$$z = \arctan\left(\frac{y_m - y}{x_m - x}\right) + \nu$$
$$\nu \sim \mathcal{N}(0, \sigma_b)$$

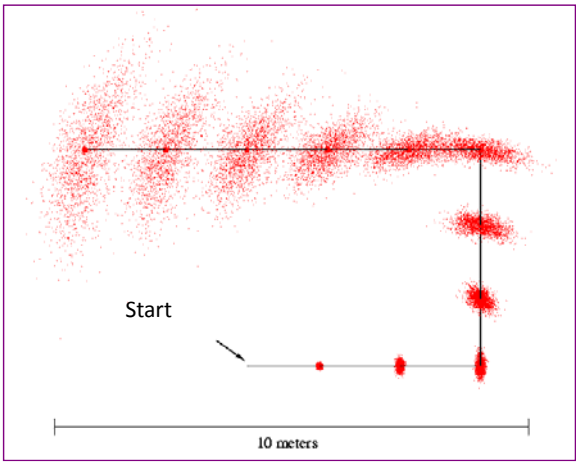


Forward vs. Inverse Sensor Models

- Forward Model
  - Common for localization
- Inverse Model
  - Useful when state less complicated than measurement
  - Used in binary occupancy grids

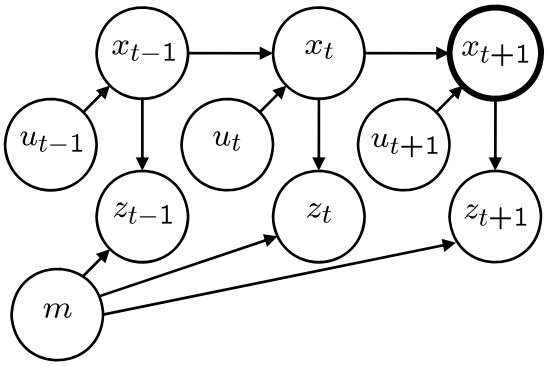


Which way did we go? Error Adds Up



Localization

- Recursive state estimation
  - General: Bayes Filter
  - Histogram (Grid Cell) Filter
  - Particle Filter
  - Gaussian (Kalman) Filters



Bayes Filter

$t++$

1. $p(x_t|u_t) = \int p(x_t|u_t, x_{t-1})p(x_{t-1})dx_{t-1}$
2. $p(x_t|z_t, u_t) = \eta p(z_t|x_t)p(x_t|u_t)$

- Start with a **prior distribution:** $p(x_0)$
- Make an observation: z_t
- Compute **posterior distribution:** $p(x_t|z_t, u_t)$
- Posterior distribution becomes the prior
- Iterate

Histogram Filter (Discrete Bayes)

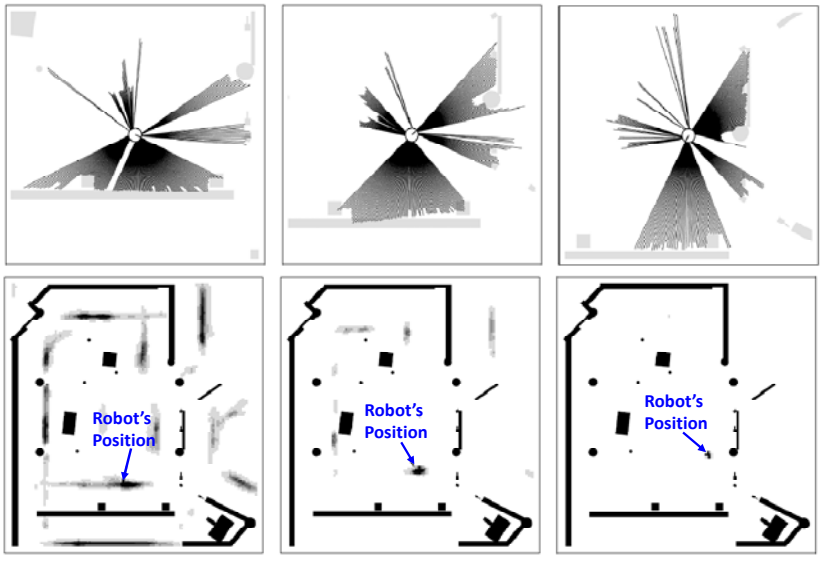
- For all possible states, iteratively compute

$$\bar{p}_{k,t} = \sum_i p(X_t = x_k | u_t, x_{t-1}) p_{i,t-1}$$

$$p_{k,t} = \eta p(z_t | X_t = x_k) \bar{p}_{k,t}$$

- Large computational expense
- Allows nonlinear, non-Gaussian models
- Implementation issues
  - Grid resolution (computationally intense)
  - Measurement independence
  - Volume of information
  - Overconfidence vs. information loss

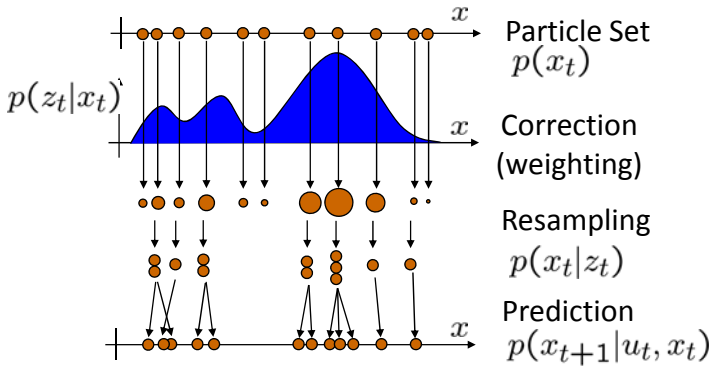
Histogram Localization



Particle Filter (Monte Carlo)

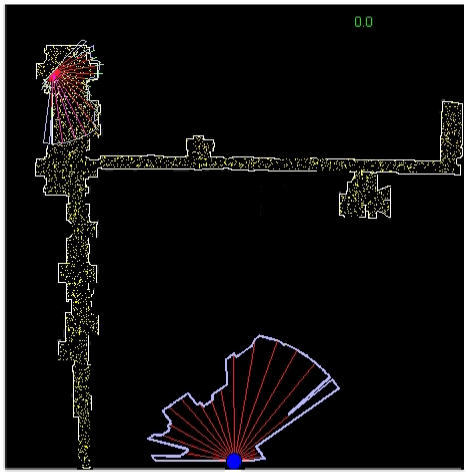
- Permits nonlinear & non-Gaussian models

… at a **large** computational expensive



Monte Carlo Localization (MCL)

- Benefits over grid
  - Continuous state representation
- Implementation issues
  - "Kidnapped robot", deprivation
  - Measurement independence
  - Volume of information
  - Overconfidence vs. information loss



Kalman Filter

- Linear system

$$\begin{aligned} x_t &= Ax_{t-1} + Bu_{t-1} + v \qquad v \sim \mathcal{N}(\vec{0}, R) \\ y_t &= Cx_t + w \qquad\qquad\qquad\quad w \sim \mathcal{N}(\vec{0}, Q) \end{aligned}$$

- Algorithm (recursive least-squares solution!)

$$\begin{aligned} \bar{\mu}_t &= A\mu_{t-1} + B\mu_t \\ \bar{\Sigma}_t &= A\Sigma_{t-1}A^T + R \\ K_t &= \Sigma_{t-1}C^T(C\Sigma C^T + Q)^{-1} \\ \mu_t &= \bar{\mu}_t + K_t\,(z_t - C\bar{\mu}_t) \\ \Sigma_t &= (I - K_tC)\Sigma_{t-1} \end{aligned}$$

Extended Kalman Filter

- Nonlinear system

$$\begin{aligned} x_t &= g(u_{t-1}, x_{t-1}) \\ y_t &= h(x_t) \end{aligned}$$

- Linear approximation

$$\begin{aligned} x_t &\approx g(\mu_{t-1}, \mu_{t-1}) + G_t(x_{t-1} - \mu_{t-1}) \\ y_t &\approx h(\bar{\mu}_t) + H_t(x_t - \bar{\mu}_t) \end{aligned}$$

Jacobians

$$\begin{cases} G_t = \dfrac{\partial g(u_{t-1}, x_{t-1})}{\partial x_{t-1}} \\ H_t = \dfrac{\partial h(x_t)}{\partial x_t} \end{cases}$$

Extended Kalman Filter (cont.)

- **Approximate** least squares solution
- Covariance estimate can be erroneous
- Basis for most modern localization systems
- Algorithm (very analogous to Kalman filter)

$$\begin{aligned}
\bar{\mu}_t &= g(u_{t-1}, \mu_{t-1}) \\
\bar{\Sigma}_t &= G_t \Sigma_{t-1} G_t^T + R_t \\
K_t &= \Sigma_{t-1} H_t^T (H_t \Sigma H_t^T + Q)^{-1} \\
\mu_t &= \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t)) \\
\Sigma_t &= (I - K_t H_t) \Sigma_{t-1}
\end{aligned}$$

Other Localization Algorithms

- Unscented Kalman Filter (Sigma Point)
  - "Linearization-free"
  - Improved estimate of covariance (usually!)
- Information Filter (equivalent to KF)
  - Beneficial when more measurements than states
  - Basis for distributed optimization

$$\Omega = \Sigma^{-1} \qquad \xi = \Sigma^{-1}\mu$$

Information Matrix — Information State

$$\begin{aligned}
\Omega_t &= \Omega_{t-1} + H_t^T Q_t^{-1} H_t \\
\xi_t &= H_t^T Q_t^{-1} (z_t - (h(\mu_{t-1}) + H_t \mu_{t-1}))
\end{aligned}$$

Linearization Example

- Two wheeled robot
  - Motion model Jacobian

$$G_t = \begin{bmatrix} 1 & 0 & \frac{v_t}{\omega_t}(\cos(\theta_t + \omega_t \Delta t) - cos(\theta_t)) \\ 0 & 1 & \frac{v_t}{\omega_t}(-\sin(\theta_t + \omega_t \Delta t) + sin(\theta_t)) \\ 0 & 0 & 1 \end{bmatrix}$$

  - Control input noise

$$R_t = V_t M_t V_t^T \quad \text{where} \quad V_t = \frac{\partial g(u_{t-1}, x_{t-1})}{\partial u_{t-1}}$$

$$V_t = \begin{bmatrix} \frac{\sin(\theta_t+\omega_t\Delta t)-\sin\theta_t}{\omega_t} & \frac{v_t(-\sin(\theta_t+\omega_t\Delta t)+\sin\theta_t)}{\omega_t^2} + \frac{v_t\cos(\theta_t+\omega_t\Delta t)\Delta t}{\omega_t} \\ \frac{-\cos(\theta_t+\omega_t\Delta t)+\cos\theta_t}{\omega_t} & \frac{v_t(\cos(\theta_t+\omega_t\Delta t)-\cos\theta_t)}{\omega_t^2} + \frac{v_t\sin(\theta_t+\omega_t\Delta t)\Delta t}{\omega_t} \\ 0 & \Delta t \end{bmatrix}$$

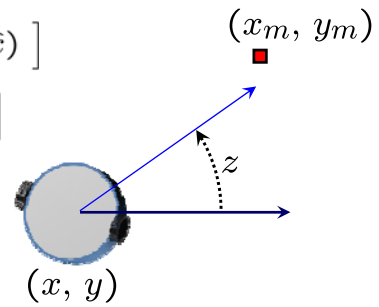
Linearization Example (cont.)

Bearing Sensor Model

$$z = \arctan\left(\frac{y_m - y}{x_m - x}\right) + \nu$$
$$\nu \sim \mathcal{N}(0, \sigma_b)$$

Sensor Jacobian

$$H = \frac{1}{\hat{r}^2}\begin{bmatrix} (y_m - \hat{y}) & -(x_m - \hat{x}) \end{bmatrix}$$
$$= \frac{1}{\hat{r}}\begin{bmatrix} \sin(\hat{\theta}_m) & -\cos(\hat{\theta}_m) \end{bmatrix}$$

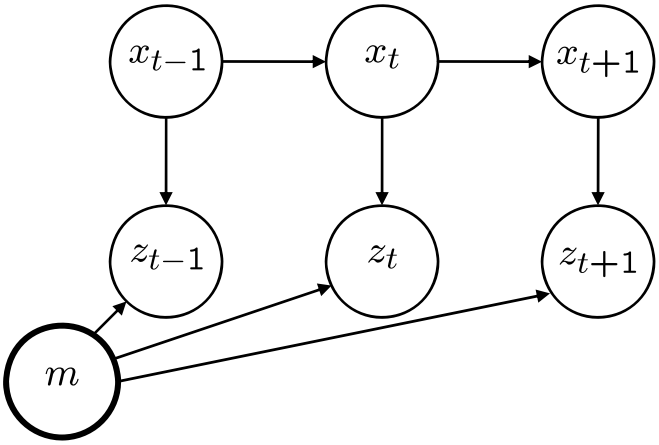


GPS + Inertial Navigation

- Sensors
  - GPS (lower altitude accuracy)
  - Accelerometer
  - Rate Gyroscope
  - Magnetometer
- Implementation
  - Outlier rejection
  - Observability requires motion
  - Sensor drift

Occupancy Grid Mapping

- Estimate the map using measurements, assume pose is known.



Binary Bayes Filter

- Estimate log odds ratio of a binary variable

$$l(x) = \log \frac{p(x)}{1 - p(x)}$$

- Uses **inverse measurement model**
- Assumes a static state
- The update for each variable is

$$l_t = l_{t-1} + \log \frac{p(x|z_t)}{1 - p(x|z_t)} - \log \frac{p(x)}{1 - p(x)}$$

Occupancy Grid with Log Odds

- Log Odds form

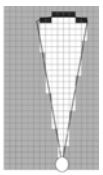
$$l_{t,i} = \log \frac{p(m_i|z_{1:t}, x_{1:t})}{1 - p(m_i|z_{1:t}, x_{1:t})} \Rightarrow p(m_i|z_{1:t}, x_{1:t}) = 1 - \frac{1}{1 + \exp(l_{t,i})}$$

- Iteratively apply to all map points in range:

$$l_{t,i} = l_{t-1,i} + \log \frac{p(m_i|z_t, x_t)}{1 - p(m_i|z_t, x_t)} - l_0$$

Inverse Sensor Model

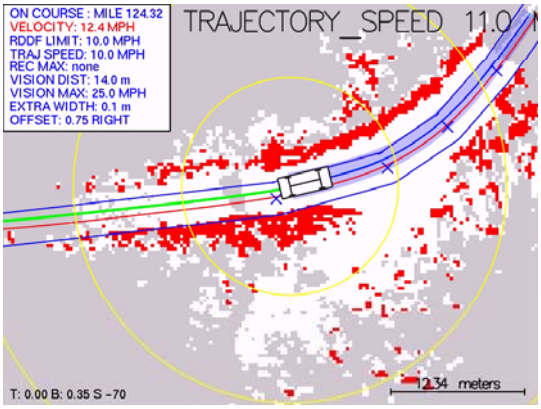
Prior Occupancy Odds: $l_0 = \log \frac{p(m_i)}{1 - p(m_i)}$



- Good numerical stability, a standard method

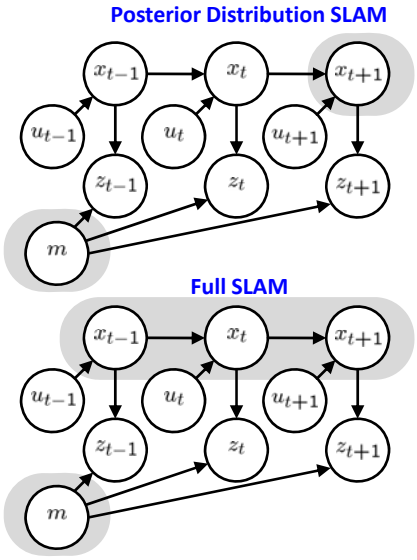
Related: DARPA Grand Challenge

- Estimated drivability (flatness)
- Used additional information
  - Time
  - Sensor vibration
- Tuned automatically



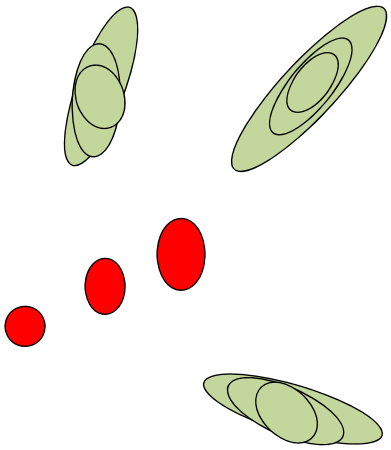
Simultaneous Localization and Mapping

- EKF SLAM
  - Feature based posterior distribution estimation
- Graph SLAM
  - Feature based full trajectory estimation
- Fast SLAM
  - Uses Monte Carlo methods



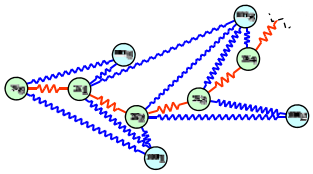
EKF SLAM

- Standard implementation tracks features
- Challenges
  - Data association
  - Divergence
  - Loop closure
  - Matrix inversion



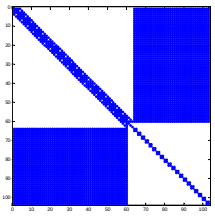
Graph SLAM

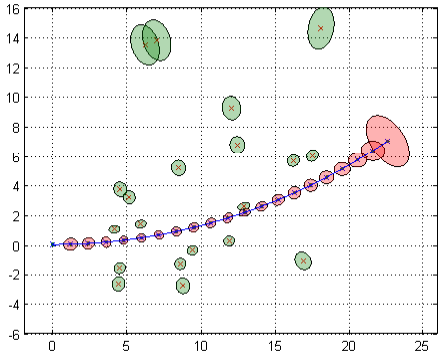
- Solves full SLAM using the information form
- Stores information links, linear storage costs



$$\Omega = \Sigma^{-1}$$

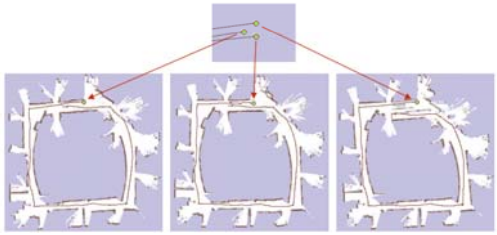
Information Matrix

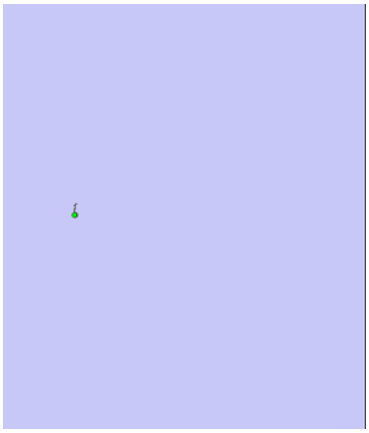




Fast SLAM

- Solves full SLAM problem
- Each particle is a trajectory
- Uses EKF feature tracking **or** occupancy grid cell





Summary

- Motion and Measurement Models
  - Statistical models approximate the system
- Localization
  - Enables estimation of current state
  - Histogram, Particle Filter, EKF (UKF, EIF, etc.)
- Occupancy Grid Mapping
  - Enables awareness of surroundings
  - Used for navigation
- SLAM
  - Matches features to enable awareness of location
  - Used for localization and mapping where global reference is unreliable/unavailable